

Web Vulnerabilities and Hardening

```
uglifier 4.1.19
lrpc 0.3.0
xmllrpc 0.3.0
lete! 31 Gemfile dependencies, 70 gems now installed.
groups test and development were not installed.
info [gemname]' to see where a bundled gem is installed.
1 message from mojo_magick:

installing MojoMagick - keepin it simple!

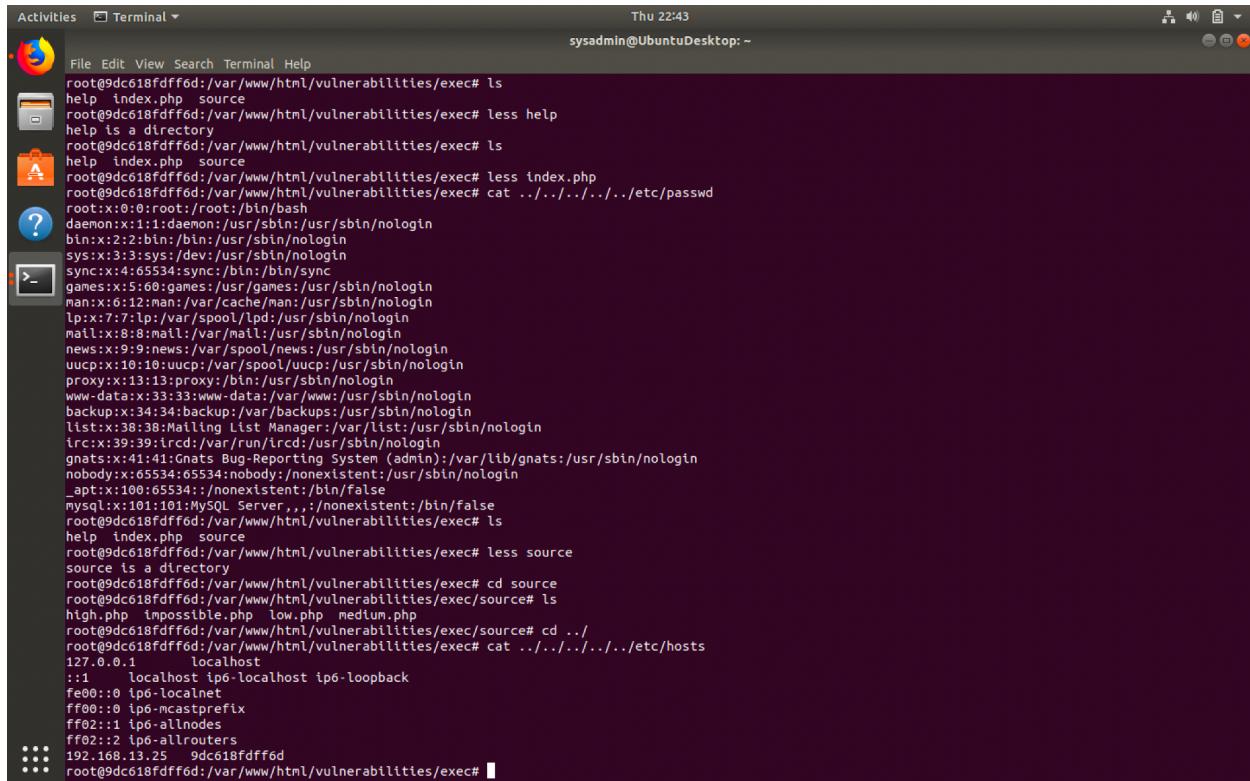
this gem work, you need a few binaries!
you've got ImageMagick available. http://imagemagick.org
to build images with text (using the "label" method) you'll need freetype and ghostscript
http://www.freetype.org and http://ghostscript.com respectively for installation info

=====
tall completed successfully!
'./beef' to launch BeEF
=====

ionPrime:~/TEST_AREA/beef/beef$ ./beef
[*] Browser Exploitation Framework (BeEF) 0.4.7.0-alpha
|_ Twit: @beefproject
|_ Site: https://beefproject.com
|_ Blog: http://blog.beefproject.com
|_ Wiki: https://github.com/beefproject/beef/wiki
[*] Project Creator: Wade Alcorn (@WadeAlcorn)
[*] BeEF is loading. Wait a few seconds...
```



1. Your wish is my command injection:



The screenshot shows a terminal window on an Ubuntu desktop environment. The title bar says "Activities Terminal". The terminal window has a dark background and displays a session where a user with root privileges is executing commands. The user runs "ls" to list files, then "less index.php" to view its source code. They then run "cat /etc/passwd" to view the password file. The user continues to run various commands like "sync", "cd", and "less source" to explore the system's configuration and files. The terminal window also shows the user navigating through the directory structure and viewing the contents of files like "hosts" and "index.php".

```
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# ls
help index.php source
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# less help
help is a directory
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# ls
help index.php source
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# less index.php
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# cat ../../../../../../etc/passwd
root:x:0:0:root:/root/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,,:/bin/false
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# ls
help index.php source
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# less source
source is a directory
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# cd source
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec/source# ls
high.php impossible.php low.php medium.php
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec/source# cd ../
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec# cat ../../../../../../etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.13.25 9dc618fdff6d
root@9dc618fdff6d:/var/www/html/vulnerabilities/exec#
```

For the mitigation for the above command injection attack, it is recommended that you harden the website.

1. Input validation code- This would ensure that only properly formed data would enter the information system. All data from untrusted sources would be subject to this, thus preventing malware or command injections. You can also use only whitelisted characters.
2. Update the Application as well as patch the system or application.
3. You can also use the principle of least privilege to only allow the application or processes the minimum requirements for the task.

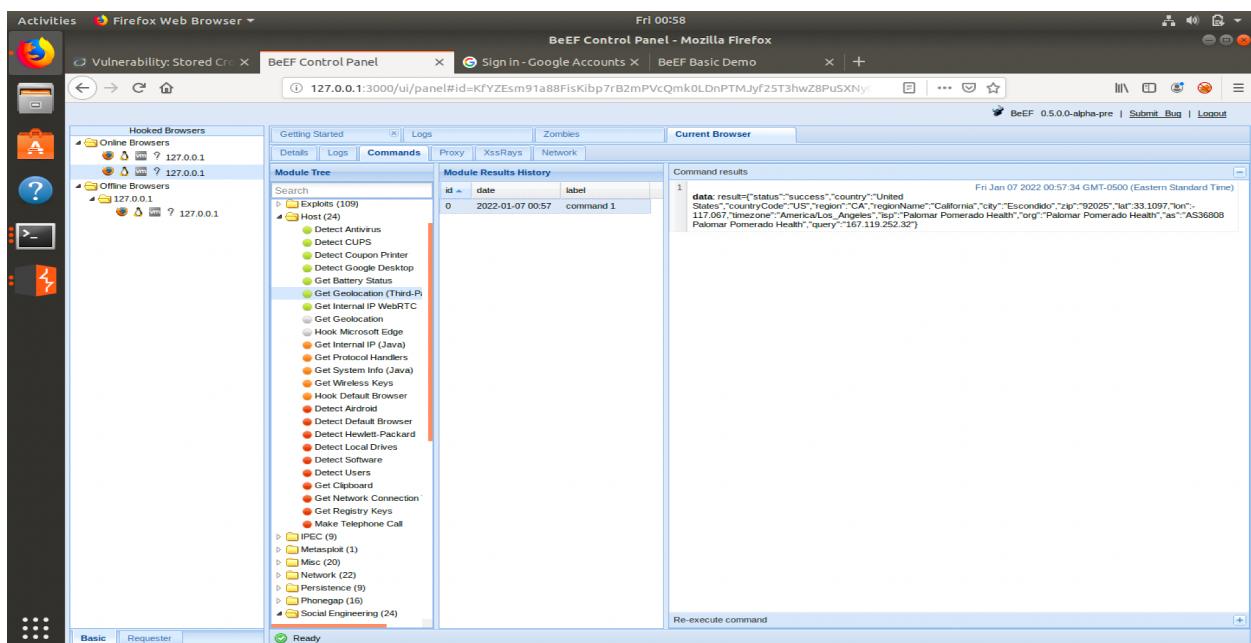
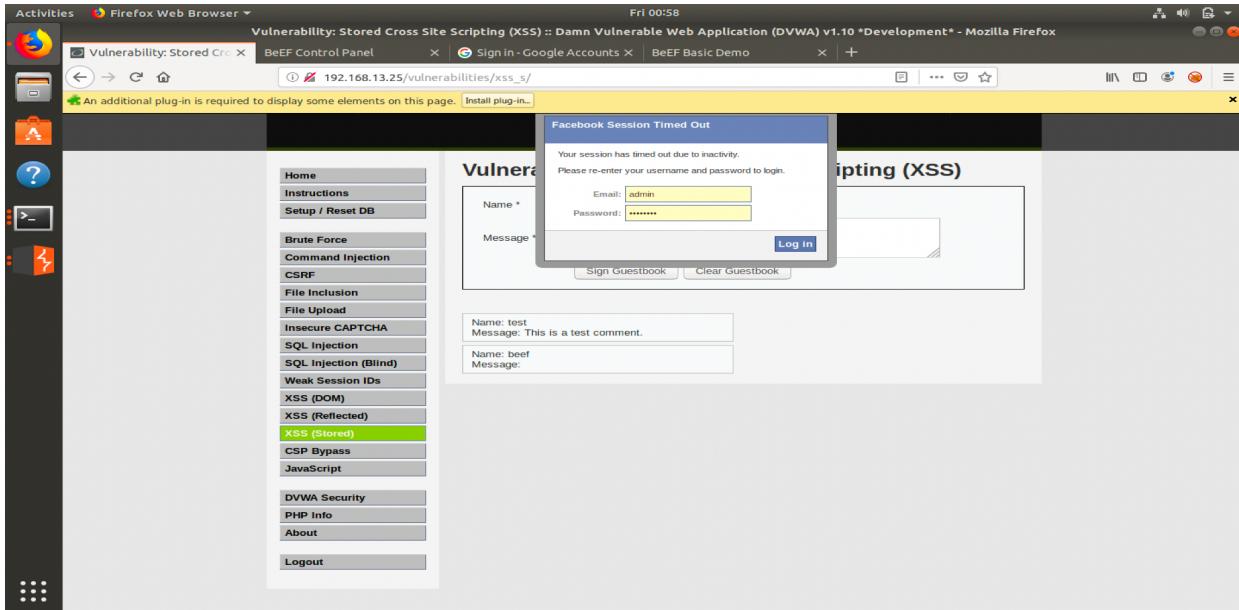
2.A Brute Force to Be Reckoned With

The screenshot shows the Burp Suite Community Edition interface. At the top, it displays 'Activities' and 'Burp Suite Community Edition'. The status bar shows 'Fri 00:31' and 'Intruder attack 2'. The main window has tabs for 'Results', 'Target', 'Positions', 'Payloads', and 'Options'. A filter bar at the top says 'Filter: Showing all items'. Below is a table with columns: Req., Payload1, Payload2, Status, Error, Timeout, Length, and Comment. The table contains 14 rows of data. Row 75 is highlighted with a red background. The 'Response' tab is selected in the Burp Suite interface. Below the interface is a web page titled '/ Broken Auth. - Insecure Login Forms /'. It has fields for 'Login:' and 'Password:', and a 'Login' button. Below the form, a message says 'Successful login! You really are Iron Man :)'. To the right of the web page are icons for LinkedIn, Twitter, and Facebook.

Req.	Payload1	Payload2	Status	Error	Timeout	Length	Comment
70	henryhacker	Courage is immortal	200			11801	
71	superman	I am Iron Man	200			11801	
72	loislane	I am Iron Man	200			11801	
73	spiderman	I am Iron Man	200			11801	
74	pennyroyal	I am Iron Man	200			11801	
75	tonystark	I am Iron Man	200			11827	
76	timstorn	I am Iron Man	200			11801	
77	peterparker	I am Iron Man	200			11801	
78	clarkkent	I am Iron Man	200			11801	
79	ironadamsmith	I am Iron Man	200			11801	
80	henryhacker	I am Iron Man	200			11801	
81	superman	His Past, Our future	200			11801	
82	loislane	His Past, Our future	200			11801	
83	spiderman	His Past, Our future	200			11801	

Mitigation strategies for a brute force attack such as this are mitigated by locking down the password attempts that can be executed preventing the use of a password cracker. If this is done the attacker will lose the chance to use a password cracker such as John the Ripper. You could also lock out authentication attempts from trusted or untrusted browsers.

3. Where's the BeEF?



Mitigation Strategies:

In order to prevent web application attacks there are many ways to harden the security to prevent as many attacks as possible. The first is to implement and redirect all HTTP traffic to HTTPS. You can also use a program such as keypass to generate and store much more and secure strong passwords. As well you can implement a web application firewall.